REMARKS

Claims 1-6 and 9-20 are now pending in the application, with claims 1, 9-11, 13 and 14 being the independent claims. Reconsideration and further examination are respectfully requested.

Objection was made to a number of different claims in the Office Action. In response, Applicant has carefully amended the claims above to address each such objection. Accordingly, withdrawal of such claim objections is respectfully requested.

Similarly, a number of claims were rejected under 35 USC § 112, second paragraph, for omitting essential structural cooperative relationships between the elements. These relationships have been clarified pursuant to the claim amendments set forth above. In addition, Applicant notes as follows.

In the above claim amendments, the purpose of the logic synthesis tool, where recited, is clarified. That is, the logic synthesis tool is used for generating a generic netlist from a RTL code description and for synthesizing the IC design. The recited constraints determine when the iterative loop between bottom-up synthesis and top-down characterization (which is performed by the logic synthesis tool under control of the generated script) is terminated. See, for example, page 84 lines 5-8 of the Specification.

Dependent claims 15-20 have been amended above to clarify that the I/O conditions and I/O constraints of the modules of the IC design are captured during the top-down characterization. This is believed to address the § 112, second paragraph, rejection with respect to those claims.

Lastly, the relationships between the generic netlist, the RTL code description, the gate-level description and the IC design are clarified in the above claim amendments. Specifically, the generic netlist is generated from the RTL code description using the synthesis tool. From the generic netlist, certain design information is obtained and is used to inform the synthesis process (or the generation of synthesis script to run the synthesis tool). This feature, in particular, is believed to be novel and non-obvious over the applied art.

The IC design may be in a variety of different formats and, in fact, the format of the IC design changes throughout the design process. Initially, the IC is designed using the RTL code. That code then may be synthesized into a gate-level description using a synthesis tool. In the present invention, prior to such synthesis the RTL code first is used to generate a generic netlist (also using the synthesis tool). That generic netlist is analyzed to determine certain information that is used in generating the synthesis scripts that are run by the synthesis tool when synthesizing the RTL code into the gate-level description. In most implementations, the first iteration of the bottom-up synthesis will be on the RTL code; then, subsequent iterations typically will perform bottom-up resynthesis on the previously generated gate-level netlist.

Based on the above claim amendments and the foregoing remarks, each of the § 112, second paragraph, rejections is believed to have been addressed. Accordingly, withdrawal of all such rejections is respectfully requested.

Claims 1-6 and 9-20 were rejected under 35 USC § 103(a) over U.S. Patent 5,812,416 (Gupte) in view of U.S. Patent 5,615,124 (Hemmi) and U.S. Patent 5,903,466

(Beausang). Withdrawal of this rejection is respectfully requested for the following reasons.

As noted above, each of the pending independent claims concerns the synthesis of an integrated circuit (IC) design from a RTL code description into a gate-level description (or the generation of synthesis script for performing such synthesis). Unlike conventional techniques in this regard, the present invention involves generation of a generic netlist and then analysis of that generic netlist for adaptively generating scripts to drive the synthesis tool. Conventional techniques, such as Gupte, have focused only on analyzing the RTL (or HDL) code in attempting to generate better synthesis scripts.

However, Applicant has recognized that merely analyzing HDL code often is not adequate and that the generation and analysis of a generic netlist often provides better results. See, for example, page 41 line 18 through page 45 line 4 of the Specification. Techniques for generic netlist analysis are discussed throughout the Specification, such as from page 81 line 20 through page 98 line 21.

Thus, each of independent claims 1, 9-11, 13 and 14 is directed to synthesizing an integrated circuit (IC) design from a RTL code description into a gate-level description (or generating synthesis script to perform such synthesis). Initially, a generic netlist is generated from the RTL code description, hardware elements are identified in the generic netlist, and then key pins are determined for each of the identified hardware elements. Design structure and design hierarchy are then extracted from the generic netlist based on the key pins and the identified hardware elements. Finally, the IC design is synthesized into a gate-level description based on the design structure and the design hierarchy extracted from the generic netlist (or script is

generated to cause the logic synthesis tool to perform such synthesis). Such synthesis includes: applying bottom-up synthesis to modules and sub-modules of the IC design, applying top-down characterization to modules and sub-modules of the IC design, and repeating such bottom-up synthesis and said top-down characterization until constraints are satisfied.

The foregoing combination of features is not disclosed or suggested by the applied art. For instance, the applied art does not disclose or suggest at least the features of: generating a generic netlist from RTL code, identifying hardware elements in such generic netlist, determining key pins for such identified hardware elements, extracting design structure and design hierarchy from such generic netlist, or using such extracted design structure and design hierarchy to synthesize an IC design into a gate-level description (or to generate script to cause the logic synthesis tool to perform such synthesis).

In this regard, it is acknowledged in the Office Action that Gupte inputs HDL code but does not explicitly disclose the use of a generic netlist. To make up for this deficiency, the Office Action cites column 1 lines 21-31 of Hemmi for the assertion that HDL code is equivalent to a "netlist". At the same time, the Office Action acknowledges that Hemmi says nothing about generic netlists, and therefore cites column 1 line 61 through column 2 line 4 of Beausang for the proposition that a generic netlist means a netlist that is technology independent, i.e., not yet correlated with a technology library. Finally, the Office Action concludes that Gupte's HDL code must be synonymous with a generic netlist.

Applicant disagrees with this line of reasoning for the following reasons. First, the cited portion of Hemmi utilizes a broad definition of HDL. See column 1 lines 21-31. That portion of Hemmi begins by noting that HDL source (which, for the reasons set forth below, is believed to be equivalent to Gupte's HDL code) is first generated by a designer to describe the desired operations of the circuitry to be constructed. Then, this portion of Hemmi goes on to note that other types of HDLs "can be considered, such as an HDL describing the connection of a transistor circuit, an HDL describing a netlist of a logic gate, and an HDL describing a hardware specification of a level higher than the circuit operation." Thus, even under Hemmi's broad definition, there are various different types of HDL (only one of which being described as a netlist, and not a generic netlist at that). Accordingly, it remains to determine which type of HDL Gupte is referencing.

In this regard, as acknowledged in the Office Action, Gupte clearly refers to "HDL code". For example, the portion of Gupte cited in the Office Action as showing the features of identifying hardware elements in a generic netlist and extracting design structure and hierarchy from a generic netlist (column 14 lines 12-22) does not in fact referred to a "generic netlist", but instead clearly refers to "HDL code".

Gupte is clear that when he refers to HDL code, he is not referring to a generic netlist. Rather, Gupte's HDL code "is a behavioral specification of the ASIC". See column 6 lines 46-47. Similarly, column 6 lines 43-45 of Gupte states, "The customer may also provide HDL source code or the foundry may help develop the code with the customer..." This latter quotation appears to confirm that Gupte's reference to HDL code is equivalent to Hemmi's reference to HDL source, i.e., the initial description of the

circuit provided by a circuit designer. Moreover, even Gupte's reference to HDL code seems to imply that it is what is conventionally understood as HDL code, i.e., the circuit designer's initial description of the circuit in a structured language.

Every single reference in Gupte appears to support this interpretation. On the other hand, nothing at all in Gupte indicates that any reference to HDL code means a generic netlist. In fact, as acknowledged in the Office Action, Gupte makes no reference whatsoever to a generic netlist.

Still further, the claims have been amended above to recite that the generic netlist is generated from a RTL code description. Certainly, nothing in Gupte indicates that the HDL code referred to in the cited portion of Gupte is a generic netlist that has been generated from a RTL code description.

Lacking this feature of the present invention, the applied art could not have disclosed or suggested any of the pending independent claims. The other claims in the application depend from these independent claims and are therefore believed to be allowable for at least the same reasons. In addition, each such dependent claim recites an additional feature of the invention that further distinguishes the invention from the applied art. Accordingly, the individual reconsideration of each on its own merits is respectfully requested.

In view of the foregoing amendments and remarks, the entire application is believed to be in condition for allowance, and an indication to that effect is respectfully requested.

If there are any fees due in connection with the filing of this paper that have not been accounted for in this paper or the accompanying papers, please charge the fees to Deposit Account No. 12-2252. If an extension of time under 37 C.F.R. 1.136 is required for the filing of this paper and is not accounted for in this paper or the accompanying papers, such an extension is requested and the fee (or any underpayment thereof) should also be charged to Deposit Account No. 12-2252. A duplicate copy of this page is enclosed for that purpose.

Respectfully submitted,

MITCHELL, SILBERBERG & KNUPP LLP

Dated: April 29, 2004

By _____
Joseph G. Swan
Registration No. 41,338

MITCHELL, SILBERBERG & KNUPP LLP
11377 West Olympic Boulevard
Los Angeles, California 90064
Telephone: (310) 312-2000
Facsimile: (310) 312-3100